УДК 004.272.43

Л.М. Сластен
Южный научный центр РАН, г. Ростов-на-Дону, Россия
lmslasten@yandex.ru

Реализация многокадровых задач на реконфигурируемых многопроцессорных вычислительных системах с ограниченной архитектурой

В статье рассматривается метод отображения многокадровых задач на реконфигурируемые вычислительные системы, архитектура которых является ограниченной. Рассматриваются особенности обеспечения взаимосвязи между кадрами, а также метод разрешения тупиковых ситуаций, возникающих в процессе отображения многокадровых задач.

Процесс разработки параллельных программ, предназначенных для реализации на многопроцессорных вычислительных системах, является сложным и требует от разработчика доскональных знаний особенностей архитектуры вычислительной системы и специальных средств для разработки и отладки. Для реализации задачи на реконфигурируемой вычислительной системе (РВС) со структурно-процедурной организацией вычислений информационный граф задачи представляется в кадровой форме [1]. Если полученный кадр невозможно разместить в РВС, то информационный граф задачи разрезается на множество подграфов (кадров), для реализации каждого из которых достаточно ресурса РВС. В результате разделения исходного кадра и формирования множества кадров задача становится многокадровой, для реализации которой необходимо отобразить каждый кадр на архитектуру РВС путем размещения вершин информационных графов кадров на элементарные процессоры и в каналы памяти и реализации множества дуг пространственной коммутационной системой. Взаимосвязь между кадрами задачи обеспечивается путем размещения информационных вершин согласно определенным правилам.

В статье рассматривается метод отображения задачи на архитектуру РВС, когда задача состоит из множества связанных между собой кадров, а архитектура реконфигурируемой многопроцессорной вычислительной системы является ограниченной.

В частном случае информационные графы многокадровой задачи могут быть независимыми, как показано на рис. 1, и в этом случае отображение задачи на архитектуру выполняется аналогично отображению задачи, содержащей единственный кадр.

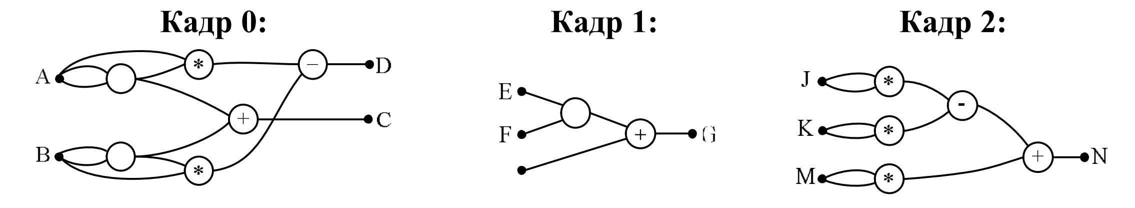


Рисунок 1 – Независимые информационные графы многокадровой задачи

В общем случае кадры многокадровой задачи взаимосвязаны, то есть имеют некоторое количество общих информационных входных или выходных вершин, как показано на рис. 2.

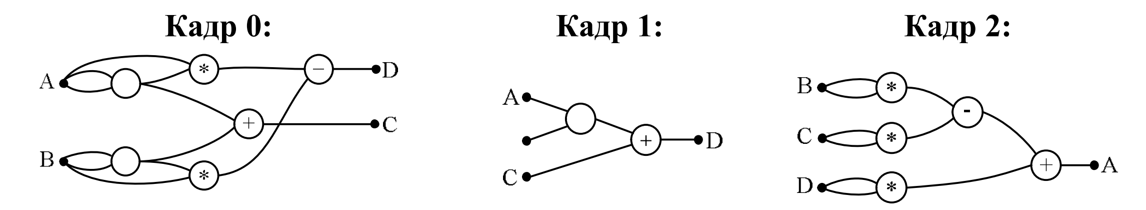


Рисунок 2 – Взаимосвязанные информационные графы многокадровой задачи

Здесь вершина A является источником данных в кадре 0 и 1 и приемником в кадре 2. Вершина B – источник данных в кадрах 0 – 2, вершина C – приемник результатов вычислений в кадре 0 и источник данных в кадрах 1 и 2. Вершина D – приемник результатов в кадре 0 и 1 и источник данных в кадре 2.

Взаимосвязь кадров задачи задается программистом при написании параллельной программы для РВС.

Реализация многокадровой задачи на РВС имеет ряд особенностей, связанных, прежде всего, с размещением информационных вершин. Существует несколько способов решения проблемы размещения информационных вершин. Один из способов заключается в том, что программист должен явно указать, как будет выполняться чтение и запись данных в каждом кадре задачи, учитывая при этом взаимосвязь между кадрами. Такой способ является очень трудоемким, поскольку количество кадров может исчисляться десятками, кадры могут иметь разнообразную структуру, а при их дальнейшем отображении возможно возникновение тупиковых ситуаций, решение которых может потребовать переразместить некоторое количество размещенных вершин, в том числе и информационных. Кроме того, при изменении архитектуры РВС программа должна быть либо откорректирована, либо переписана заново.

Другой способ заключается в том, что программист указывает, какие именно источники и приемники являются общими для взаимосвязанных кадров многокадровой задачи, но не указывает, в каких каналах памяти они будут расположены. При выполнении процедуры отображения многокадровой задачи на архитектуру РВС информационные вершины, соответствующие источникам и приемникам кадров, размещаются в первую очередь, после чего выполняется автоматическое размещение операционных вершин графов кадров задачи, совмещенное с трассировкой информационных каналов. Исследования показали, что выполнять размещение информационных вершин отдельно от размещения операционных вершин и трассировки информационных каналов нецелесообразно, потому что информационные и операционные вершины тесно связаны между собой. При использовании данного способа отображения многокадровой задачи разрешение тупиковых ситуаций, которые возможны при отображении задач, является очень трудоемкой процедурой.

Для решения проблемы реализации многокадровых задач на РВС был разработан новый метод, который рассматривается в данной статье. В отличие от задачи, содержащей единственный кадр, при реализации многокадровой задачи на РВС необходимо реализовать ряд дополнительных преобразований.

В общем случае кадры многокадровой задачи имеют различную структуру, информационные графы кадров различаются количеством вершин и связей. Очевидно, что отображение многокадровой задачи следует начинать с обработки самого

сложного кадра, граф которого имеет наибольшее количество вершин. Кадры задачи упорядочиваются по убыванию степени сложности и образуют последовательность, которая задает порядок обработки.

Согласно методу, рассмотренному в [2], операционные вершины информационного графа объединяются в клики, к которым в процессе размещения добавляются неразмещенные информационные вершины, которые сократят или не увеличат количество связей клики с вершинами, не принадлежащими ей.

При размещении клики, содержащей информационные вершины, следует учитывать, что соответствующие ей данные должны быть сохранены для всех кадров, графам которых принадлежит данная вершина. На рис. 2 представлены информационные графы взаимосвязанных кадров и показаны общие информационные вершины. Здесь, например, все информационные вершины содержатся в графе каждого из трех кадров. Однако необходимо учитывать, что выполнение кадров, графам которых принадлежит информационная вершина, может чередоваться с кадрами, графы которых не содержат данную информационную вершину. Для сохранения данных, соответствующих вершине, необходимо определить номера кадров, в которых область памяти в канале, выделенная для данных вершины, не может быть использована.

Для этого из номеров кадров, графам которых принадлежит вершина, выбираются минимальный и максимальный элементы: *MinCN* и *MaxCN* соответственно. Тогда для каждого кадра *i*, где *i = MinCN,…, MaxCN*, данные в канале, в который размещена информационная вершина, сохраняются, что иллюстрируют рисунки 3а) и б).

Здесь вершина *E* принадлежит графам кадров с номерами *1*, *i*, *j*, поэтому ее данные должны сохраняться для каждого кадра *k = 1, …, j*. В свою очередь вершина *F* принадлежит графам кадров с номерами *i–1*, *j + 1*, *NC*, поэтому ее данные должны сохраняться для каждого кадра *k = i – 1, …, NC*.

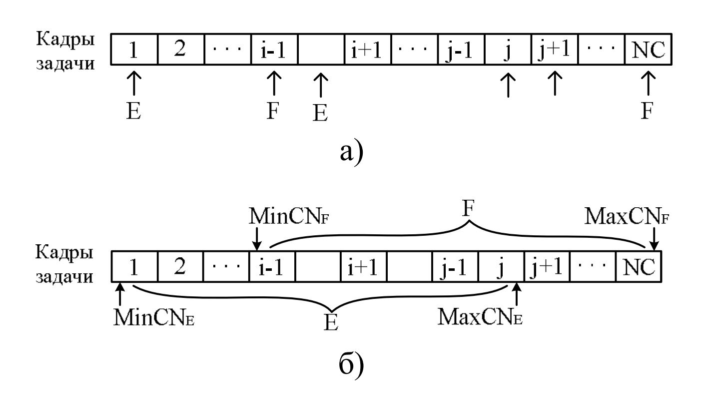


Рисунок 3 – Номера кадров, для которых информационные вершины сохраняют свои значения

При размещении информационной вершины в канал памяти для каждого кадра *i = MinCN, …, MaxCN* объем свободной памяти в канале необходимо декрементировать на величину, равную объему данных, соответствующих информационной вершине. Таким образом, информационную вершину графа обрабатываемого кадра можно разместить в канале памяти при условии, что объем данных информационной вершины не превышает объем свободной памяти в канале. В случае, когда объем данных информационной вершины больше максимально допустимого объема любого из каналов памяти, многокадровая задача не может быть реализована на данной РВС. Размещение информационной вершины в канал памяти иллюстрирует рис. 4а).

Для информационных вершин, как и для операционных, может потребоваться переразмещение в случае, когда отображение обрабатываемого кадра заходит в тупик. При переразмещении информационной вершины необходимо освободить канал, в котором она размещена, и изменить объем свободной памяти в канале, инкрементировав его на величину, равную объему данных, соответствующих данной информационной вершине. Эти действия выполняются для каждого кадра, информационному графу которых принадлежит переразмещаемая информационная вершина. Удаление информационной вершины из канала памяти иллюстрирует рис. 4б).

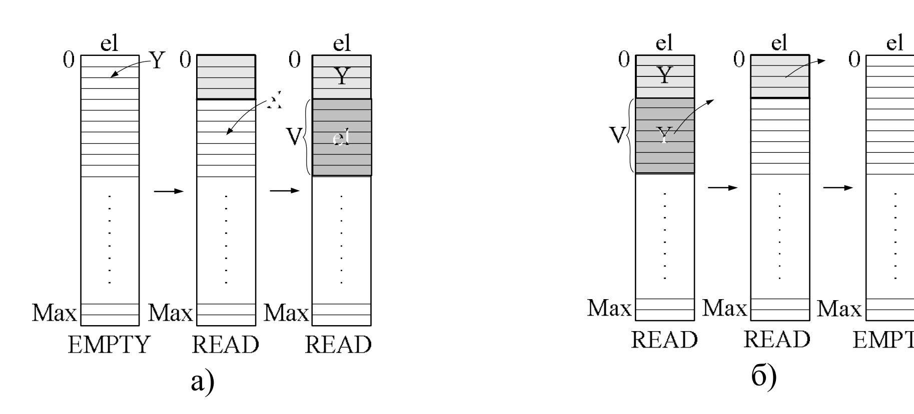


Рисунок 4 – Размещение и удаление информационной вершины

Процедура переразмещения выполняется, когда возникает тупиковая ситуация, то есть размещаемую клику невозможно разместить или для размещаемой клики невозможно сформировать одну из трасс [3]. Разрешение тупиковой ситуации заключается в том, чтобы отменить размещение предыдущей размещенной клики, переразместив ее, и снова попытаться разместить клику, при которой возникал тупик. Для этого последовательность размещения клик сохраняется и представляет собой кортеж Z, каждый элемент которого z_i, $i = 1, \ldots, NZ$, является парой значений: номером размещенной клики и номером кадра, при обработке которого данная клика размещена. Здесь NZ – общее количество размещаемых клик многокадровой задачи. Для доступа к элементам кортежа Z имеются два указателя: iz – указатель размещаемых клик и cz – указатель размещенных клик (рис. 5).

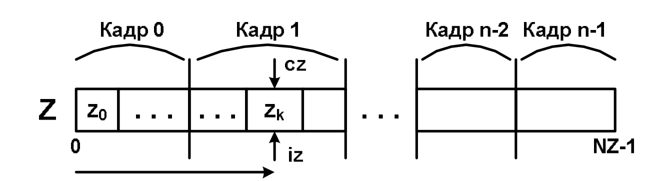


Рисунок 5 – Кортеж размещенных клик многокадровой задачи

Когда происходит переразмещение клики, т.е. когда последняя размещенная клика становится неразмещенной и размещается повторно, указатель cz декрементируется, а указатель iz не изменяется. Процедура отката показана на рис. 6.

Переразмещение возможно, только когда значение указателя cz не равно нулю, то есть когда существует, по крайней мере, одна размещенная клика. Если возникает необходимость отката, то после переразмещения одной или нескольких клик последовательность выбора клик для повторного размещения должна быть прежней, т.е. повторно должна быть размещена именно та клика, из-за которой совершался откат.

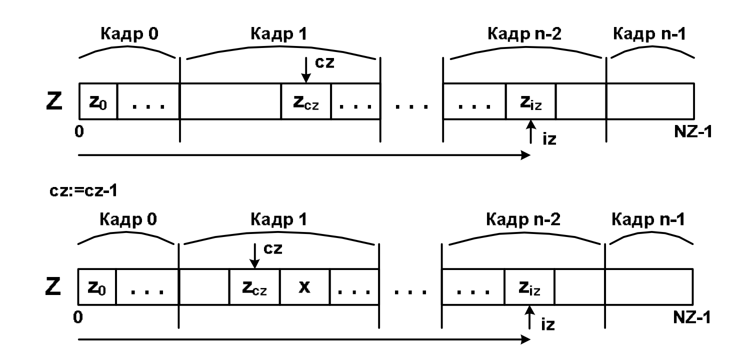


Рисунок 6 – Процедура отката

Поскольку задача состоит из множества кадров, необходимо учитывать, что информационные вершины, которые являются общими для нескольких кадров, должны помещаться в последовательность размещенных клик только один раз. Поэтому при выборе нового обрабатываемого кадра выполняется анализ состояния информационных вершин выбранного кадра. Если информационная вершина принадлежит множеству кадров и при этом не является размещенной, то есть не содержится ни в одной из клик последовательности *Z*, то она должна быть размещена при обработке данного кадра. В противном случае информационная вершина была размещена при обработке одного из предыдущих кадров и она не может добавляться в клики обрабатываемого кадра.

При выполнении отката и переразмещения одной из ранее размещенных клик для многокадровой задачи учитывается то, что возможен переход к предыдущему обработанному кадру, если он существует.

При реализации однокадровой задачи на РВС для разрешения возможных тупиковых ситуаций откат выполняется последовательно к предыдущей размещенной клике, счетчик *cz* декрементируется на единицу, после чего сразу же выполняется переразмещение обрабатываемой клики. В ряде случаев последовательный откат на одну клику и ее переразмещение не могут привести к успешному размещению тупиковой ситуации, поскольку причины возникновения тупика иные и не связаны с данной переразмещаемой кликой, а клика, которая явилась реальной причиной возникновения тупиковой ситуации, соответствует значению счетчика $cz\text{-}k$, где $k > 1$.

Последовательный перебор клик с уменьшением счетчика *cz* на единицу и попытки их переразмещения увеличивают время отображения информационного графа на архитектуру РВС. Поскольку задача может состоять из множества кадров, то последовательность размещенных клик резко увеличивается, в результате чего время отображения задачи становится очень большим.

В некоторых случаях можно избежать последовательных откатов с единичным шагом и многочисленных переразмещений путем анализа секции-источника и секции-приемника для той трассы, которую создать не удалось. Если для связи между вершиной текущей размещаемой клики и вершиной, которая принадлежит одной из клик, размещенных ранее, не удалось сформировать трассу, возможны две причины отказа трассировки:

1) заблокирована секция, в которую предполагается разместить выбранную клику;

2) заблокирована секция, в которой содержится ранее размещенная клика.

В этом случае достаточно перебрать варианты размещения размещаемой клики, и если ни один вариант не подходит, то следует анализировать варианты размещения для тех ранее размещенных клик, для связи с вершинами которых не удается

сформировать трассы. Для этого определяется клика, которой принадлежит ранее размещенная вершина, для связи с которой трассировка оказалась неудачна, и номер клики *k* в последовательности размещенных клик. Если ранее размещенная клика содержит вершины, которые принадлежат информационному графу исключительно текущего кадра, то выполняется откат на *k* элементов последовательности *Z* и переразмещение данной клики (рис. 7).

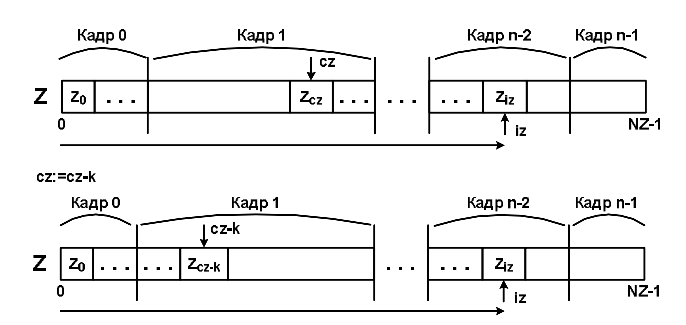


Рисунок 7 – Процедура отката на *k* шагов в пределах обрабатываемого кадра

Если ранее размещенная клика содержит вершины, которые принадлежат информационным графам нескольких кадров, то, возможно, потребуется отменить размещение ранее размещенных клик обрабатываемого кадра, а также одного или нескольких кадров, размещенных ранее (рис. 8).

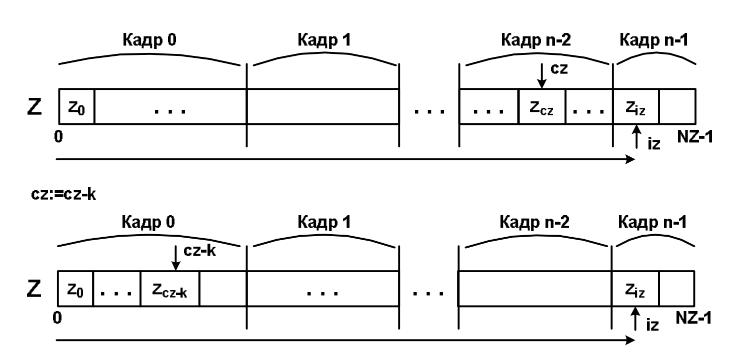


Рисунок 8 – Процедура отката на *k* шагов с отменой размещения ранее размещенных кадров

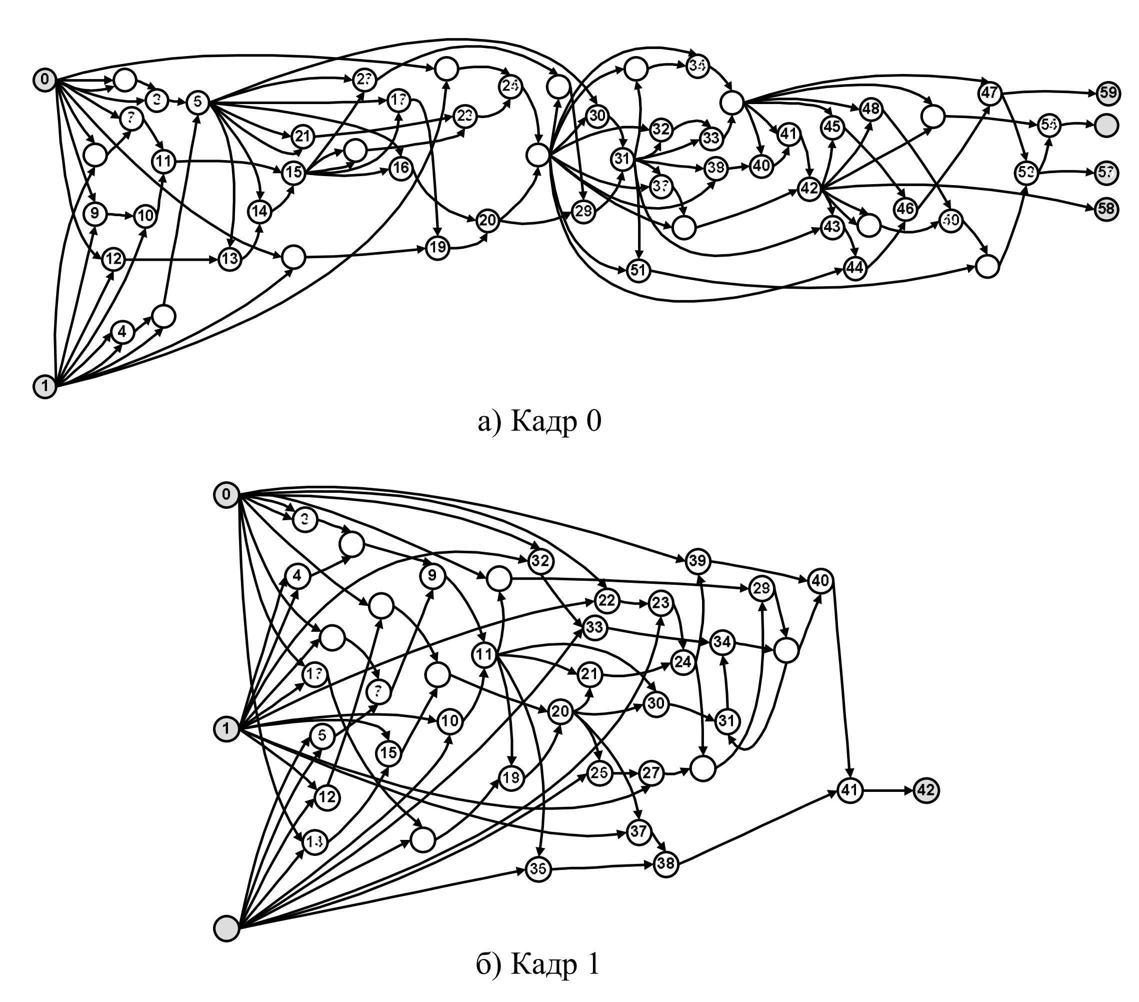
Здесь при выполнении отката отменяется размещение клик кадра 0 и *n-2* и полностью отменяется размещение графов кадров от 1 до *n-3*. Клика с номером *cz-k* будет переразмещена, после чего будет выполнено повторное размещение неразмещенных клик кадра 0, будут полностью отображены информационные графы кадров от 1 до *n-3*, а также будет размещена клика с номером *cz*.

Процедура отката и переразмещения выполняется до тех пор, пока либо кадр успешно отображается на архитектуру РВС, либо достигается самая первая клика в последовательности размещенных клик.

Если в результате переразмещения была достигнута самая первая размещенная клика и выбраны все возможные варианты ее размещения, но ни один не оказался приемлемым, то в таком случае решение задачи размещения не найдено. Делается вывод, что данная многокадровая задача не может быть реализована на архитектуре РВС, и необходимо изменить структуру кадра, в котором возникла неразрешимая тупиковая ситуация и отказ при отображении многокадровой задачи, разбив его на несколько кадров.

Отображение графа обрабатываемого кадра завершено, если все неразмещенные вершины графа кадра размещены. Счетчик кадров задачи инкрементируется, выбирается новый обрабатываемый кадр, если он существует, и размещение вершин выполняется снова. В противном случае все графы кадров задачи отображены на архитектуру РВС, и отображение многокадровой задачи завершено.

Ниже, на рис. 9, приведен пример задачи, состоящей из двух взаимосвязанных кадров, и результаты ее отображения с использованием разработанных алгоритмов на РВС с различными видами архитектуры.



а) Кадр 0

б) Кадр 1

Рисунок 9 – Информационные графы кадров многокадровой задачи

Здесь, в кадре 0, вершины с номерами 0 и 1 по условиям задачи являются входными информационными вершинами и соответствуют операции чтения. Вершина 0 используется только в нулевом кадре. Вершина 1 – общая для обоих кадров задачи. Входной информационной вершине 1 нулевого кадра соответствует выходная информационная вершина 42 первого кадра, то есть в канале памяти существует некоторое множество адресов, общее для кадров 0 и 1 и соответствующее вершинам 1 и 42. В нулевом кадре из данной области памяти выполняется операция чтения данных, а в первом кадре – операция записи данных.

В кадре 0 вершины с номерами 64 – 67 по условиям задачи являются выходными информационными вершинами и соответствуют операции записи. Вершина 67 используется только в нулевом кадре. Вершины 64 – 66 – общие для обоих кадров

задачи. Выходным информационным вершинам 64 – 66 нулевого кадра соответствуют входные информационные вершины 0 – 2 первого кадра, то есть для каждой из пары общих вершин кадров 0 и 1 в каналах памяти существует множество адресов, общее для кадров 0 и 1. В нулевом кадре в данные области памяти выполняется операция записи данных, а в первом кадре – операция чтения данных.

На рисунке 10а) – г) представлены результаты отображения взаимосвязанных кадров многокадровой задачи на РВС с различными видами архитектуры.

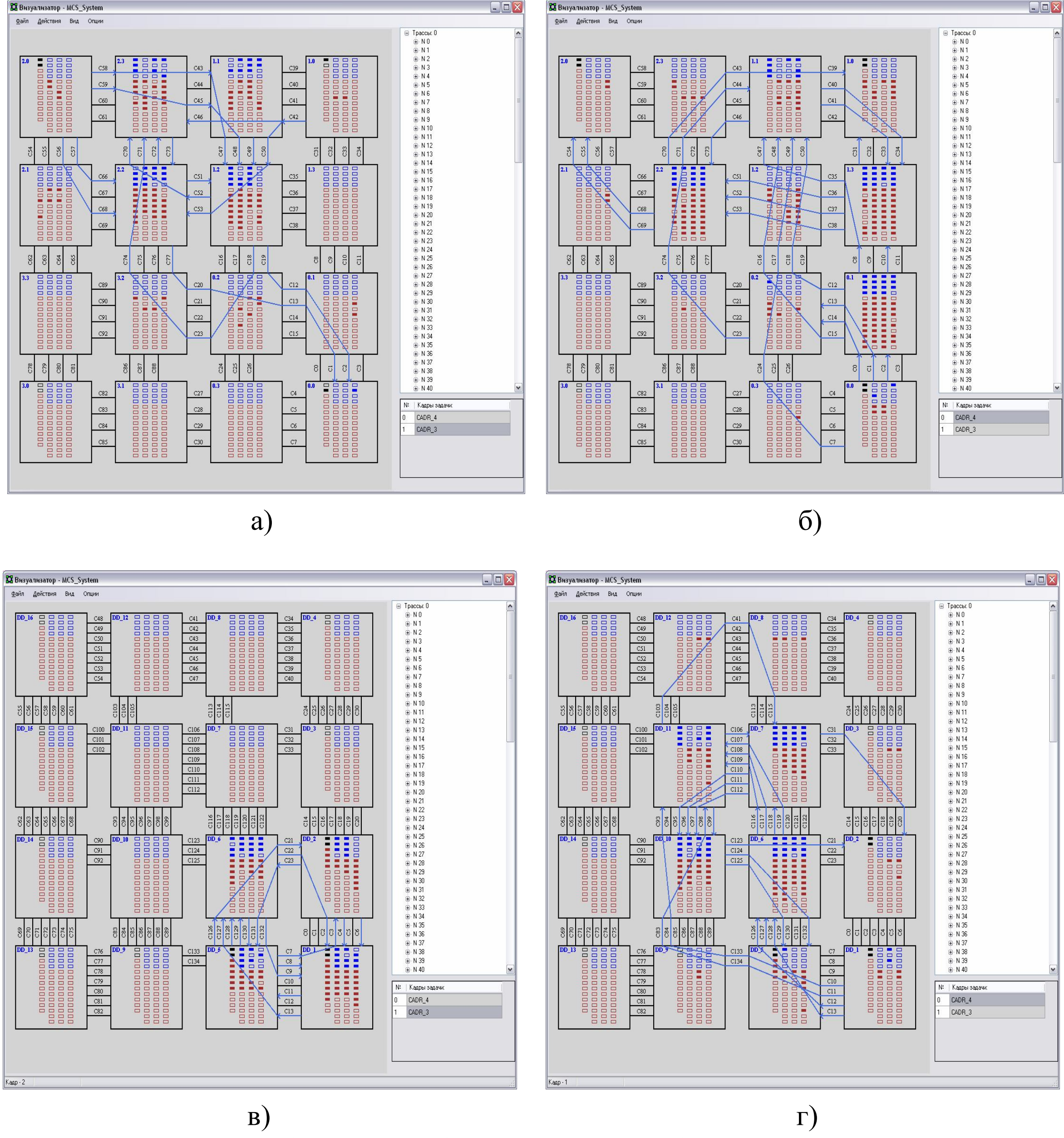


Рисунок 10 – Отображение информационных графов кадров многокадровой задачи на РВС с различными видами архитектуры

Результаты исследований показали, что рассмотренный метод отображения многокадровых задач на архитектуру РВС позволяет значительно сократить трудоемкость разработки и отладки прикладных параллельных программ, состоящих из множества

взаимосвязанных кадров. Программа-синтезатор, реализованная на основе рассмотренного метода, выполняет отображение прикладной параллельной многокадровой задачи на архитектуру РВС различных видов автоматически, избавляя прикладного программиста от необходимости отслеживать размещение информационных вершин, обеспечивающих взаимосвязь кадров, а также от детального знания всех особенностей архитектуры РВС. Рассмотренный метод поиска и разрешения тупиковых ситуаций позволяет сократить время отображения многокадровой задачи на архитектуру РВС за счет сокращения количества переразмещений вершин информационных графов.

Литература

1. Каляев И.А. Реконфигурируемые мультиконвейерные вычислительные структуры / [Каляев И.А., Левин И.И., Семерников Е.А., Шмойлов В.И.] – Ростов-на-Дону : Изд-во ЮНЦ РАН, 2008. – 320 с.
2. Сластен Л.М. Отображение графа задачи на архитектуру реконфигурируемой вычислительной системы с ограниченной коммутационной структурой / Л.М. Сластен // Системы и средства искусственного интеллекта (ССИИ-2008) : материалы Международной научной молодежной школы, (22 – 27 сентября 2008, пос. Кацивели, АР Крым). – Донецк : ИПИИ «Наука і освіта», 2008. – С. 111-116.
3. Самоделкова Е.А. Метод формирования трасс при отображении графа задачи на архитектуру реконфигурируемой вычислительной системы с ограниченной коммутационной структурой / Е.А. Самоделкова, Л.М. Сластен // Системы и средства искусственного интеллекта (ССИИ-2008) : материалы Международной научной молодежной школы, (22 – 27 сентября 2008, пос. Кацивели, АР Крым). – Донецк : ИПИИ «Наука і освіта», 2008. – С. 106-111.

Л.М. Сластьон

Реалізація багатокадрових задач на реконфігурованих багатопроцесорних обчислювальних системах з обмеженою архітектурою

У статті розглядається метод відображення багатокадрових задач на реконфігуровані обчислювальні системи, архітектура яких є обмеженою. Розглядаються особливості забезпечення взаємозв’язку між кадрами, а також метод розв’язання безвихідних ситуацій, що виникають у процесі відображення багатокадрових задач.

Статья поступила в редакцию 04.07.2009.